REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
24-06-2008	Final Report	20-Oct-2007 - 19-Jun-2008

4. TITLE AND SUBTITLE	
Cognitive Particles Final Report	5a. CONTRACT NUMBER
	5b. GRANT NUMBER W911NF-08-C-0005
	5c. PROGRAM ELEMENT NUMBER 7620AK
6. AUTHORS Darby Grande, Webb Stacy, Georgiy Levchuk, Nathan Schurr	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES	8. PERFORMING ORGANIZATION REPORT NUMBER
Aptima, Inc. 12 Gill St. Suite 1400 Woburn, MS 01801 -	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO 11. SPONSOR/MONITOR'S REPORT NUMBER(S) 53379-MS-DRP.1

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; Federal purpose rights

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

Cognitive Particles is a step toward realizing the ability to develop autonomous components (particles) that are capable of coordinating to come together and form a desired object. Automated shape assembly and disassembly from a collection of particles would allow incredible resource availability and flexibility in domains ranging from the highly specialized and time-critical (soldiers in the field or medical technicians in the operating room) to the more mundane. In this work, we first developed theoretical concepts supporting the development and coordination of autonomous shape-forming particles. Building on this foundation, we constructed a 3-dimentional particle simulation testbed to enable experimentation regarding

15. SUBJECT TERMS

Programmable Matter, Self-forming Materials, Automated Shape Assembly, Command and Control, C2 Networks, Distributed Control

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Georgiy Levchuk
U	U	U	SAR		19b. TELEPHONE NUMBER 781-496-2467

Report Title

Cognitive Particles Final Report

ABSTRACT

Cognitive Particles is a step toward realizing the ability to develop autonomous components (particles) that are capable of coordinating to come together and form a desired object. Automated shape assembly and disassembly from a collection of particles would allow incredible resource availability and flexibility in domains ranging from the highly specialized and time-critical (soldiers in the field or medical technicians in the operating room) to the more mundane. In this work, we first developed theoretical concepts supporting the development and coordination of autonomous shape-forming particles. Building on this foundation, we constructed a 3-dimenstional particle simulation testbed to enable experimentation regarding hypotheses about the processes and structures required for automated shape assembly.

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations:

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Number of Manuscripts:

Number of Inventions:

Graduate Students

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Faculty Supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED

FTE Equivalent:
Total Number:

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: ......

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:......

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:......

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale): ......

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering: ......

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense ......

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields: ......

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

Inventions (DD882)

www.aptima.com
Woburn MA • Washington DC

Security Classification: Unclassified

Final Progress Report

Submission Date: June 24, 2008

Cognitive Particles

#1383

Reporting Period: 10/20/2007 – 06/19/2008

Prepared for:

ARO Technical Representative: Dr. David M. Stepp
U.S. Army Research Office
ATTN: AMSRD-ARL-RO-EM
P.O. Box 12211
Research Triangle Park, NC 27709-2211

Phone: (919) 549-4329
Email: david.m.stepp@us.army.mil

DARPA Technical Representative: Mitch Zakin
DARPA
3701 North Fairfax Drive
Arlington, VA, 22203-1714

Phone: (703) 248-1509
Email: mitchell.zakin@darpa.mil

Prepared by:

Dr. Georgiy Levchuk, PI
Dr. Darby Grande, Dr. Webb Stacy, Dr. Nathan Schurr
Aptima, Inc.
12 Gill Street, Suite 1400
Woburn, MA 01801
Tel. 781.496.2467

Abstract

Cognitive Particles is a step toward realizing the ability to develop autonomous components (particles) that are capable of coordinating to come together and form a desired object. Automated shape assembly and disassembly from a collection of particles would allow incredible resource availability and flexibility in domains ranging from the highly specialized and time-critical (soldiers in the field or medical technicians in the operating room) to the more mundane. In this work, we first developed theoretical concepts supporting the development and coordination of autonomous shape-forming particles. Building on this foundation, we constructed a 3-dimenstional particle simulation testbed to enable experimentation regarding hypotheses about the processes and structures required for automated shape assembly.

Key Terms

- Programmable Matter
- Self-forming Materials
- Automated Shape Assembly
- Command and Control
- C2 Networks
- Distributed Control

Table of Contents

List of Figures

Statement of Problem Studied

The challenge motivating the Cognitive Particles project is to develop autonomous components (particles) that are capable of coordinating to come together and form an arbitrary object. This research is part of a larger emerging field is often referred to as Programmable Matter because of its goal of creating a substance that can be programmed to change its material properties (e.g., shape, density, or even color). Most current research in Programmable Matter focuses on the lower level issues of how to enable its component particles to successfully complete actions such as move, communicate, and join with adjacent particles. Alternatively, the Cognitive Particles project focuses on the coordination challenges that will arise once these actions are made possible. In order to investigate these future challenges, we concentrate on three key questions:

- *What are the coordination challenges of automated shape assembly?*
- *What is necessary for the automated object assembly planning and execution?*
- *What are important metrics of object formation, and how does the object plan and execution affect them?*

Cognitive Particles has begun to answer each of these questions, which has led to accomplishments in the areas of (1) Design and development of a Cognitive Particle Testbed, (2) Cognitive Particles theory development, and (3) Cognitive Particles metrics and results. These are all readily extensible to the Programmable Matter field. Furthermore, this initial investigation into a broad and burgeoning field of study has provided insights into numerous promising directions for future research. These accomplishments, lessons learned, and future directions are detailed below.

Summary of Important Results

Under the Cognitive Particles project, we have worked to design and develop the theory and technology required to realize and exercise a simulation testbed for automated shape assembly from fundamental particles. This work first involved developing the concepts necessary to hypothesize about and analyze the process of automated shape assembly. These are formalized in the report of accomplishments below. Additionally, we designed and implemented the simulation testbed by combining the software capabilities of a structured database, a Java-based model prototyping environment, a 3-D graphics visualization engine, and an engine for real-time physics simulation.

The important results from Cognitive Particles come from both the theory development tasks as well as the work to implement the simulation testbed. The theory of automated shape assembly and disassembly led us through questions of what it means to have different kinds of particles together in a bucket, whether the particles should be controlled in centralized or decentralized manner, how the particles should communicate, and how do we introduce energy into the system. On the other hand, the construction of the simulation testbed allowed us to ask questions about the limitations and benefits of various particle and shape representation choices, the level of visualization required for an operator to understand simulation trials, how explicitly do

collisions and links between particles need to be represented, their computational burdens, and the initialization conditions appropriate for a group of particles.

One of the clear issues that we identified in Cognitive Particles is the importance of understanding the tradeoffs between what needs to be determined online during object formation, and what needs to be planned offline in advance. To reduce the memory burden on individual particles, important calculations such as the decomposition of the desired final shape into subshapes for construction (and the determination of the required number of "leaders" for these subshapes), should be performed offline if it is to be done in advance of the start of shape construction.

We then found that we could (and sometimes should) augment this master shape plan (computed offline) with additional information beyond simply the decomposition into subshapes. First, the color (or material) requirements for different parts of the shape could be added to achieve the mechanical or visual effects. Second, the order in which the subshapes must be constructed and fused can be developed to ensure that assembly of a complex shape is feasible. This information could be developed offline, and we conducted this analysis for several shape types.

During the online real-time object execution, we found several possibilities for enhancing the construction process: multiple leaders can coordinate to build and fuse parts of a larger shape in order to achieve construction parallelization for efficiency; dynamic, real-time allocation of particles to "roles" in the shape--where in the shape they should end up--allows us to specify rules for how the particles determine where they should be based on location, particle type, the control structure, etc.; the addition of extra or varying forces such as shaking the particles' container, gravity, and magnetic forces between particles can affect the speed of particle assembly, change the density of particles in different areas of the container, and allow better mixing of the particles throughout the container; and finally, the use of dissemination of color to test communication patterns and network integrity provides a conveniently visual analysis of these structures.

Report of Project Accomplishments

In the following sections, we provide the details of the work under the Cognitive Particles project.

Developing the Theory of Automated Shape Assembly: Definition of Concepts

In this subsection, we define the main concepts that we used to represent the objects and shapes, their content, the physical environment, and planning processes to form the shapes needed.

Defining the Objective

The objective of the user is to construct a certain **physical object of interest** from the set of elementary components. The *object description*, which is developed by the user, specifies the properties of the object in the form of its shape and kinetics. The *shape definition* might be of descriptive nature, but must be translated into a topological specification of how elementary

components may form this shape. Such specification, described in detail in the next section, can be either defined by the user, or can be automatically derived using *quantization* of the shape form, given that the assembly system can match the description of the form with known shape. For example, the user may desire to build a "ball"; the system must understand that the ball is of spherical form with a surface equally distanced from the center of the shape.

The *kinetics definition* for the object may include the desired properties that the user may wish to obtain. For example, the user may specify a desire for the ball to stay intact under a heavy pressure and to have a high bounce capability. Such definitions can be translated into topological properties of the component elements of the shape and their connections. An example might be a "baseball" that has multiple layers with different properties (cork, rubber, and mixture of the two, with liner components) designed to achieve desired resistance and weight properties.

Particles: The Elementary Components

The *elementary components* that can be used to build a shape must be defined. As a construction company may use bricks or wood components to build a house, the user may specify the type of elementary components available to conduct the assembly. In our work, we have used the *particles* as the elementary components with the shape of a small 3-dimentional cube. The size of the particles was fixed to be the same, while the color could be varied for the visualization purposes. The particles were assumed to be able to connect to each other along their faces – so that any two connected particles are well aligned along the corresponding faces (Figure 1).

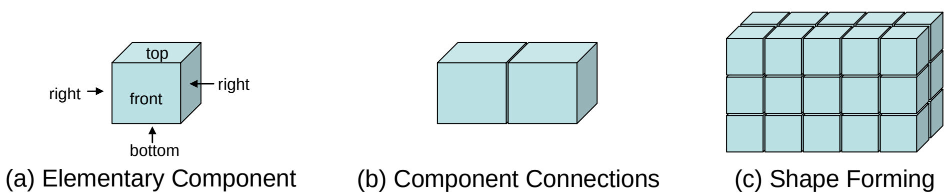


Figure 1: Elementary Components: Particles and Their Connections

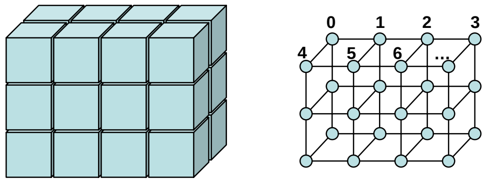
Without loss of generality, we considered the elementary particles with *homogeneous physical properties* – that is, we assumed that all elementary particles are made of (or require) the same material to be manufactured, and thus are equivalent in the physical world. We made a similar assumption about the connections among particles, assuming that a single type of connection exists. This can be expended to particles with heterogeneous properties, as is needed for construction of a baseball as described above, as well as heterogeneous links, for example rigid joints, springy links, or rag-doll-like connections.

Shape Structure Specification

The structure of the desired shape is topologically defined as a 3-D graph, where the nodes are particles that must constitute the shape and links are joints between these particles. The nodes in this graph are indexed with integer values. The links then carry information about the connecting node indexes and faces of the corresponding cube particles. Figure 2 shows an example of the shape representation. We call this topological construct a *node-link specification* of the desired shape.

As we are dealing with well-aligned cube particles and homogeneous links, the geo-spatial information (e.g., location and orientation) of the particle in the shape is not needed. Moreover,

this information is redundant, because we assume the user does not care what orientation the shape will be manufactured at.



(a) 3-D Shape (b) 3-D Graph

Link ID	Node From	Node To	Face From	Face To
0	0	1	front	rear
1	0	4	right	Left
2	1	2	front	rear
3	1	5	right	Left
...	...	...	...	...

(c) Shape Specification

Figure 2: Node-Link Specification of Desired Shape

Quantitatively, and more generically, the shape topology can be specified as an *attributed network*, where the nodes represent particles, links represent the joints among them, and the attributes on nodes and links describe the profiles (or properties) of the particles and their connections. In our work, we only assumed that the connections have attribute information – defined using faces of the particles the connection is supposed to join. Such network can be defined using a triplet $G=(V,E,A)$, where $V=\{1,...,|V|\}$ are nodes corresponding to components of the shape, E is a set of links between them, and A is a set of attributes on links and nodes determining the properties of the particles and their connections ($A=\|a_{ij}\|$, where a_{ii} is attributes vector for node i and a_{ij} is attribute vector for link between nodes i and j). This description can define both directed and undirected shape specifications (in case of undirected specification the attributed matrix A is symmetric).

We define the physical (current) particle network using the variables $G_C=(V_C,E_C,A_C)$ and a desired shape using $G_D=(V_D,E_D,A_D)$. Current physical structure changes over time, as the shape is being built or disassembled. The desired shape structure remains constant according to the node-link specification of the shape topology. Note that the user might specify *multiple objects* as the objectives for the manufacturing, where only a single object must be manufactured. Such specification may be needed when the user can be satisfied with obtaining any of the several objects with various degrees, and the manufacturing process has cost-benefit tradeoff. For example, the user might desire to build pliers or scissors, and while pliers would match the most to the needs, the construction of the scissors may be simpler and this object would satisfy the requirements to a certain degree. The system then must intelligently weigh in different values to come up with the specific node-link spec to be executed during assembly.

The Execution Intelligence: Command and Control Network

In our work, we rely on the assembly being conducted with the help of particles that have imbedded intelligence. Availability of such particles is limited, but they bring the value of distributed shape assembly that is not available in a centralized construction processes. As the result, we distinguish two types of particles (Figure 3):

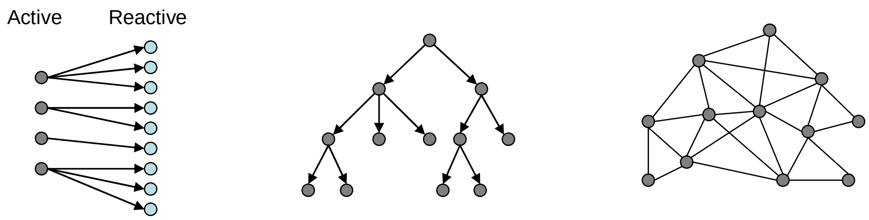
- **Reactive particles:** these are standard particles (sometimes referred to as *resources*) with limited memory and no intelligence, and can be "told" to create connections with other particles.

- **Active particles:** these are particles that have memory and intelligence. Sometimes referred to as *commanders*, active particles can decide about the instructions that must be executed to construct a shape, communicate information, request information, send commands to other active particles, and send instructions to reactive particles.

Both active and reactive particles possess ability to move in the environment and create joints (connections) with other particles.

Active and reactive particles form the organization, referred to as *command and control* assembly organization. It has the following elements:

- **Control network:** we define an assignment of reactive particles to active particles. An active particle can send instructions only to those reactive particles that it is assigned in a control network. A control network can be defined using a variable $c_{ij}=1$ if the active particle i is assigned reactive particle j (otherwise $c_{ij}=0$). Only one active particle can *control* the reactive particle – that is, $\sum_i c_{ij}=1$. Essentially, the control network is a *bipartite graph*.
- **Command network:** we define a command hierarchy using the variables $h_{ij}=1$ if active particle i is a commander of active particle j (otherwise $h_{ij}=0$). In the hierarchical command, active particle can have only a single commander, - that is, $\sum_i h_{ij}\le 1$, - with a single "top commander" of the command network (for this node we will have $\sum_i h_{ij}=0$).
- **Communication and information flow network:** we define the ability of active particles to exchange information with other active particles using variables $n_{ij}=1$ if active particle i can send information to active particle j (otherwise $n_{ij}=0$).



(a) Control Net (b) Command Net (c) Communication Net

Figure 3: Command and Control (C2) Assembly Organization

A communication network may depend on the geo-spatial distribution of the particles – that is, on the ability of the particles to transmit the information (e.g., using wireless peer-to-peer communication the other particles may pose obstacles and the distance may change the ability to communicate information). On the other hand, command and control networks are defined more as "roles" – that is, the command and control relationships should not be changing significantly over time unless the organization is adapting to the environment – see discussion in the "Future Directions" section.

Different organizations (variables $<C,H,N>=<\|c_{ij}\|,\|h_{ij}\|,\|n_{ij}\|>$) would allow different control processes, trading off more distributed execution with higher levels of control over this process. This will result in different assembly execution times and even accuracy for some of the shapes. As the result, to achieve higher degree of shape assembly correctness and decrease the assembly time, the particle organization must be matched to (tailored to, congruent with) the shape node-link specification and corresponding shape temporal plan (defined in next subsection).

The Execution Process: Instructions for Roles Assignment and Joint Construction

The nodes V_D in a shape node-link specification $G_D=(V_D,E_D,A_D)$ are essentially a set of roles that must be filled by the physical particles. Any particle – active or reactive – may fill the role of the desired shape. As the result, we need to find a *role mapping matrix* $S=\|s_{kj}\|_{k\in V_C,\, j\in V_D}$, where variables s_{ij} define the particle-to-role assignment. That is, $s_{ij}=1$ if particle i is mapped to (is assigned a role of) the particle j in the desired shape network. When the roles are selected, the connections (joints) must be built. That is, for the two particles $k,m\in V_C$, if they are assigned the roles $i,j\in V_D$ (that is, $s_{ki}=s_{mj}=1$), then there must be a joint between k,m if $e_{ij}^D=1$ with attributes a_{ij}^D and no joint if $e_{ij}^D=0$. When the joints are constructed successfully, we have: $e_{km}^C=e_{ij}^D, a_{km}^C=a_{ij}^D$. For a completely successfully built shape, we can write:

$$e_{km}^C=\sum_{ij}s_{ki}s_{mj}e_{ij}^D, a_{km}^C=\sum_{ij}s_{ki}s_{mj}a_{ij}^D .$$

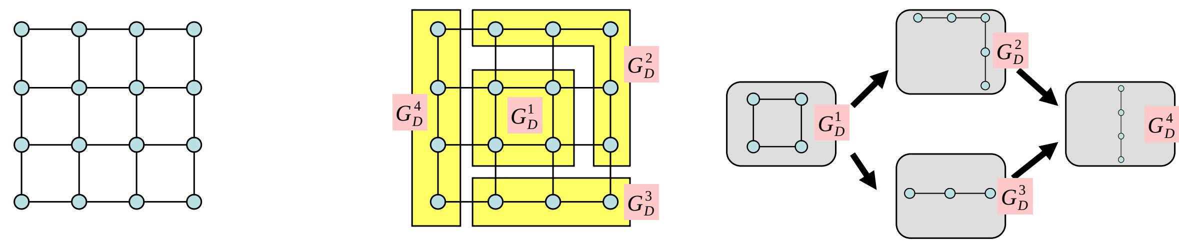
As the result, two instructions must be generated: *role assignment* and *joint construction*. The first instruction helps the active particle maintain knowledge about the roles of its own and its controlled reactive particles. The second instruction is needed to execute connections by reactive particles.

Execution Planning: The Shape Assembly Plan

The desired shape structure $G_D=(V_D,E_D,A_D)$ will be built by the C2 assembly organization. To utilize the ability of the active particles to generate instructions and supervise the shape execution process in parallel, we can create a *shape assembly plan* that has two major components (Figure 4):

- **Shape decomposition** defined as multiple subsets V_D^s of node set V_D ($\bigcup_s V_D^s=V_D; V_D^s\cap V_D^r=\varnothing$).Note that accordingly, we can define a subshape s as $G_D^s=(V_D^s,E_D^s,A_D^s)$, where E_D^s are links among nodes in V_D^s and A_D^s are corresponding attributes of nodes and links. We can define the subshape s using variables u_{si}, where $u_{si}=1$ if the node $i\in V_D$ is in the subshape s, that is $i\in V_D^s$, and $u_{si}=0$ otherwise. Then, $V_D^s=\{i\in V_D:u_{si}=1\}, E_D^s=\{(i,j)\in E_D:u_{si}\cdot u_{sj}=1\}$.

- **Shape temporal plan** defined as a precedence graph of subshapes G_D^s. This can be done using variables $p_{sr}=1$ if G_D^s must be constructed before starting G_D^r (and $p_{sr}=0$ if no such restrictions exist).



(a) Shape Node-Link Spec (b) Subshape Decomposition (c) Shape Plan

Figure 4: Example of Shape Assembly Plan

In Figure 4, a shape is decomposed into four subshapes. Figure 4c shows an example of the shape temporal plan, where precedence constraints define the temporal ordering between building the subshapes. In this example, the shape assembly will start with subshape G_D^1, then subshapes G_D^2 and G_D^3 could be assembled in parallel, and then a subshape G_D^4 will complete the shape construction. Such temporal constraints must be tracked over time, with subshapes assembly activated only when all its predecessor subshapes in the shape temporal plan have been constructed successfully. This monitoring is done by the active particles in the assembly C2 organization. We can assign the responsibilities of the subshape activation to the active particles that are supervisors of the active particles building the constituent subshapes (a subshape and all its predecessors). When active particle completes its subshape, it reports this status to the supervising active particle, which then determines if the next subshapes could be activated. For the example in Figure 4, Figure 5 shows the hierarchy of active particles and the activation responsibilities.

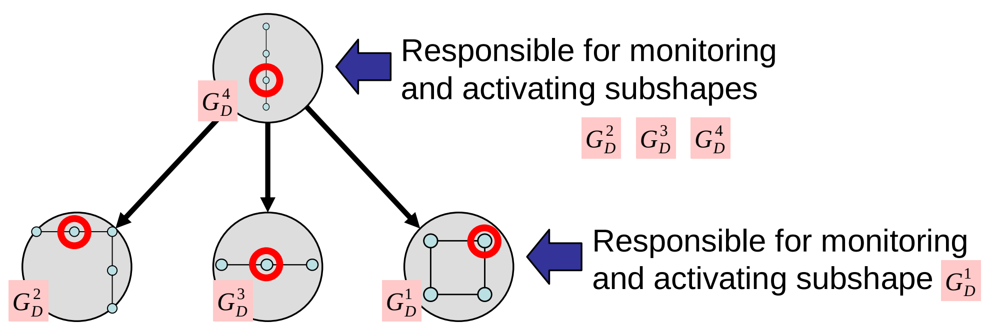


Figure 5: Example of Subshape-to-Active Particle Allocation and Activation Responsibility Assignment (the roles of active particles in the subshape are selected by active particles; in this figure, these roles are marked with red circles)

A shape decomposition is used to assign the subshapes to active particles. We can define this assignment using the variables $x_{si}=1$ if the subshape G_D^s is assigned to active particle i and $x_{si}=0$ otherwise. In our work, we assigned only a single subshape to active particle, so that $\sum_i x_{si}=1, \sum_s x_{si}\le 1$.

Assembling the Shape: Summary of Steps

In this section, we summarize the steps used for the shape assembly, starting with the shape definition and ending with the physical component assembly.

Assembly Planning

According to the above, the following *assembly planning* is performed off-line to create the *object plan*. This process is defined in the following steps (Figure 6):

Step 1: Perform quantization of the shape to develop a 3-D component model

Step 2: Extract 3-D Graph from the component 3-D model and node-link specification

Step 3: Conduct shape decomposition and develop shape temporal plan

Step 4: Design the C2 organization to support the shape assembly, including command, control, and communication networks

Step 5: Assign the shape assembly plan elements (the subshapes) and the subshape activation responsibilities to the active particles in C2 organization for the assembly execution

As the result of the assembly planning, the C2 organization is ready to start execute the shape assembly. Note that some of the steps above can be performed jointly to improve the efficiency of the product solutions (e.g., steps 4 and 5).

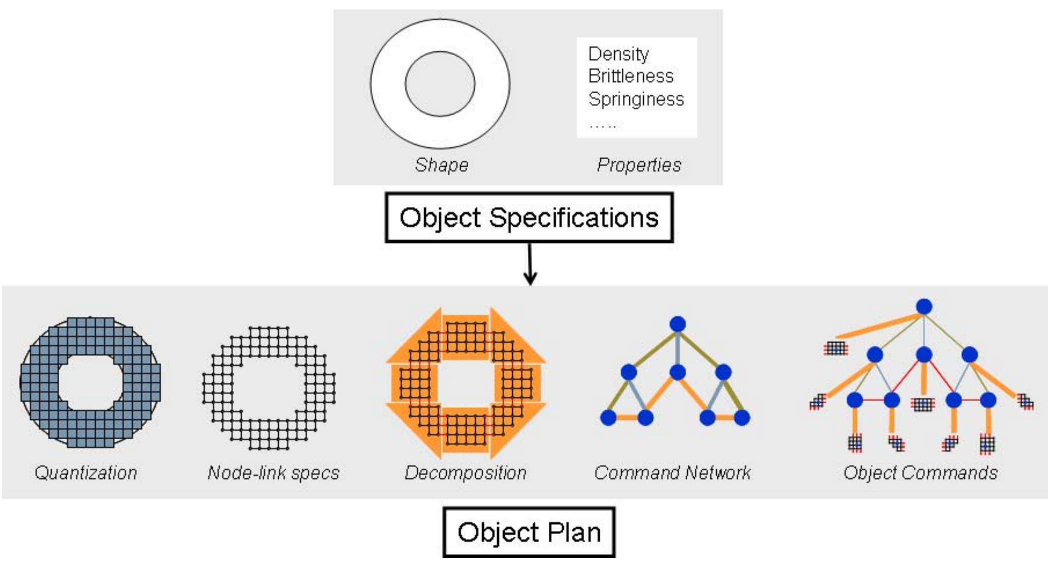


Figure 6: Example of Shape Assembly Plan (off-line process)

Assembly Execution

The following *assembly execution* is performed on-line to create the *shape*. This process is performed by active particles and defined in the following steps (Figure 7):

Step 1: Assign unfilled roles in the subplan to the active particles

Step 2: Determine the remaining set of unfilled roles

Step 3: Allocate the remaining unfilled roles to a subset of available and unused controlled reactive particles.

Step 4: Generate connection (joint) instructions based on mismatch between the current state of the shape and desired shape, and send these instructions to reactive particles

Step 5: Update the subshape unfilled roles

Step 6: When the subshape is finished, report to the active particle monitoring its success; if an active particle receives a report of subshape completion, update the successor subshapes in the shape temporal plan and activate the subshapes if possible.

Step 7: When the subshape is finished, connect/fuse this subshape with existing (already constructed) subshapes

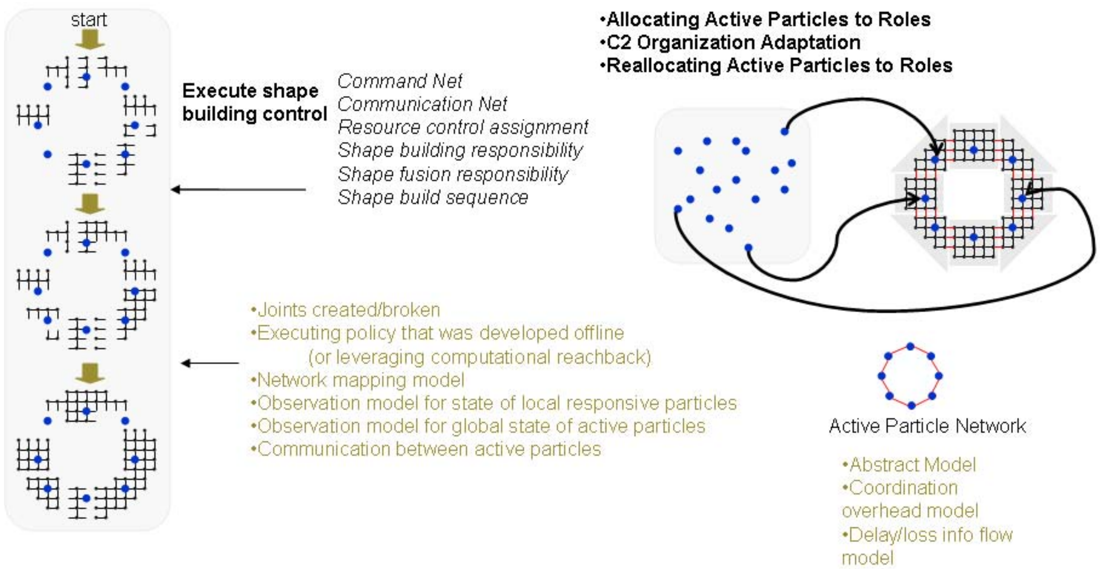


Figure 7: Example of Shape Assembly Execution (on-line process)

Assembly Models

In this section, we describe specific models we used for planning and execution of the shape assembly.

Shape Decomposition and Control Network Design

Due to the computational complexity, the shape decomposition and control network design is performed offline. The decomposition of the object node-link specification $G_D = (V_D, E_D, A_D)$ into a set of subshapes $G_D^s = (V_D^s, E_D^s, A_D^s)$ can be obtained manually, but the complexity of decisions about such decomposition for large-component object prevents the user from constructing the composition in real time. Instead, we investigated the automated decomposition approaches that trade-off three main variables:

- **Internal workload of active particles:** as the subshape assembly must be executed by the active particles, the workload of the subshape assembly (e.g., the number of nodes in

the subshape and the joints/links that must be constructed) becomes important. This is due to the limit on the memory and computational power that active particles may possess. More formally, the internal workload for the subshape $G_D^s = (V_D^s, E_D^s, A_D^s)$ is defined as the weighted sum of the nodes and links in the shape: $w^I(s) = w_{link}\sum_{i,j} u_{si}u_{sj}e_{ij}^D + w_{node}\sum_i u_{si}$, where w_{link}, w_{node} are the loads of building single link and monitoring single node respectively.

- **External workload of active particles:** the subshapes must be "fused" together to form the shape. Such fusion must be conducted by coordinating between the active particles that control reactive particles that must have joints constructed between them. More formally, the external workload for the subshape $G_D^s = (V_D^s, E_D^s, A_D^s)$ is defined as the weighted sum of the links with other subshapes: $w^E(s) = w_{link}\sum_{r \neq s}\sum_{i,j} u_{si}u_{rj}e_{ij}^D$
- **Complexity of subshape sequencing:** the subshapes must be sequenced to enable the fusion to occur. Some decompositions result in efficient parallelization of the subshape construction process, while other decompositions may result in the sequential shape building and as the result higher cost and delays of the construction process. The subshape sequencing is addressed in the next subsection describing shape temporal plan design.

In the above, the notion of "workload" is introduced to model the coordination among active particles:

- *Internal coordination* – to control the assembly of subshape managed by an active particle; and
- *External coordination* – to fuse its subshape with subshapes constructed by other active particles.

The shape decomposition results in the total workload of the subshape assemblies equal to $w(s) = w^E(s) + w^I(s) = w_{link}\sum_r\sum_{i,j} u_{si}u_{rj}e_{ij}^D + w_{node}\sum_i u_{si}$. We can then define the objectives or constraints for shape decomposition based on *balancing* or *constraining* these workloads. Such balancing or constraining is required due to limited memory and computational power at the active particles.

As the result, the shape decomposition can be posed as an optimization problem: we need to find a clustering of the nodes of the shape that achieves some optimization of the inter- and intra-cluster properties. One example of such problem is to minimize the squared sum of subshape workloads $\min_{u_{si}}\sum_s w^2(s)$, while another example is to maximize the entropy $\max_{u_{si}}\sum_s \frac{w(s)}{\sum_r w(r)}\log\frac{w(s)}{\sum_r w(r)}$. Both problem formulations would result in balancing the workloads of subshapes assembly. This optimization can be carried out using non-linear optimization techniques, with barrier functions and Lagrangian relaxation providing the most efficient solutions. In our work, we used *Tabu search algorithm* that iteratively found a sub-optimal subshapes using the "manipulations" of the solution to move to another solution. Many

manipulations are possible; instead, we focus only on a limited manipulation set that allows simple update and objective/constraints recomputation:

1. Change of assignment of node to a different subshape
2. Swap assignments of 2 nodes between their subshapes
3. Crossover in assignment vector of two subshapes, which results in swapping of the assignments of multiple nodes

The Tabu algorithm maintains a list of assignments that must not be changed for some period of time. It is also allowing (with a small probability) the manipulations of the solution to occur that result in degradation in the value of objective function, which allows the search to avoid local optimums.

Shape Temporal Plan Design

In this section, we describe how the shape temporal plan and subshape fusion temporal constraints can be generated. *Due to the computational complexity, the shape temporal planning design is performed offline.* First, we note that the requirement for sequencing the shape construction comes from the situations in which one subshape is "inside" another subshape. Three examples of this situation, with four subshapes color-coded, are shown in Figure 8. In both examples, subshape A is inside subshape B. In Figures 8a and 8b, the assembly of subshape A does not have to precede assembly of subshape B, because both subshapes can be constructed in parallel and then subshape A can "slide into" subshape B (see Figure 9). This is not the case with example of Figure 8c, in which if the subshapes A and B are constructed in parallel, A cannot be fit into B and thus B would have to have a part of it disassembled. We thus require, to avoid unnecessary disassemblies, to construct the subshape A first, and then continue constructing the "surface" of subshape B by first creating the joints of particles of A and B. That is, the "fusion" of A and B must start before the construction of subshape B. One of the ways to do this is to construct a single "external" link (joint) from a node in A and node in B, and then proceed iteratively (the iterative construction approach is described in more details in "Iterative Role Selection" section).

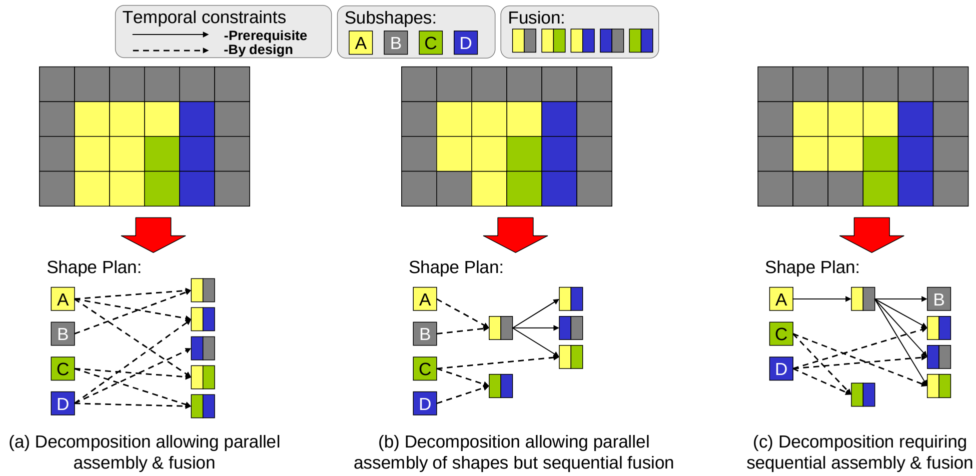


Figure 8: Two Examples of Alternative Shape decompositions requiring Different Shape Plan Temporal Constraints

Also note that there are no requirements to sequence the subshape fusion in Figure 8a, because all fused shapes can similarly “slide into” the other subshapes. This is not the case for decomposition examples of Figures 8b and 8c: if B is fused with D, or A is fused with D, the other subshapes cannot be moved to fit the structure.

Only the example in Figure 8c requires the sequencing of the shape construction. Whenever this happens, we require the fusion joints to be constructed first. While in other circumstances some sequencing of subshape fusion is not necessary, we simplify the planning process by introducing “by design” temporal constraints. In our model, if otherwise not specified, the fusion will occur after the individual subshapes have been assembled. That is, we only constrain the necessary temporal fusion sequences, and allow other fusion to occur opportunistically.

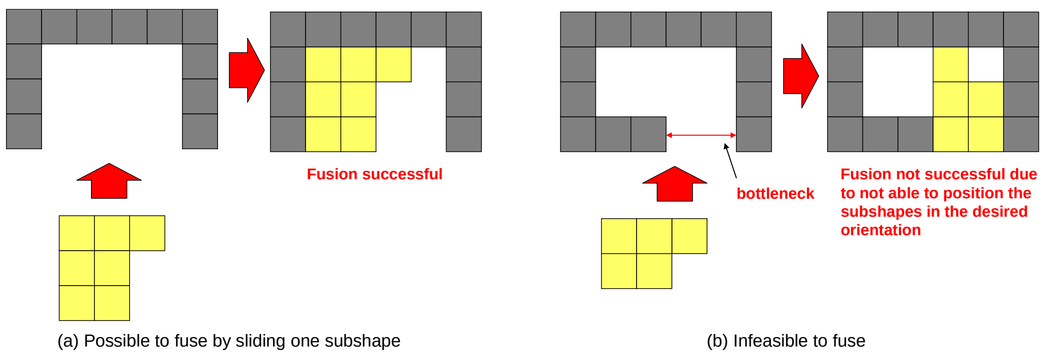


Figure 9: Two example of the fusion of two subshapes

Subshape Assembly Execution: Iterative Role Selection

The subshapes are assembled on-line: active particles conduct necessary computations, updates, generate decisions, and communicate them to each other and to reactive particles. Both types of particles then execute the instructions by performing *move* and *connect* actions.

When the assembly of subshape allocated to active particle is activated, its construction proceeds iteratively as follows. Initially, an active particle selects a ***role*** (a node in the subshape specification) for itself. Then, the algorithm iteratively determines the next available roles to fill, and selects reactive particles for these roles. The role is said to be *available* if it must be connected to a particle that is *finished*, - that is, there exists a joint in the node-link specification of the subshape between this role node and the role node of the finished particle. A particle is said to be finished if is has a role selected and all the joints have been formed. The status of roles is updated, and this process is repeated for the next set of “available” roles in the shape plan. An example of this process in 2-dimensional space is shown in Figure 10. While all available particles are considered at every iteration, not all of them are fulfilled and assembled.

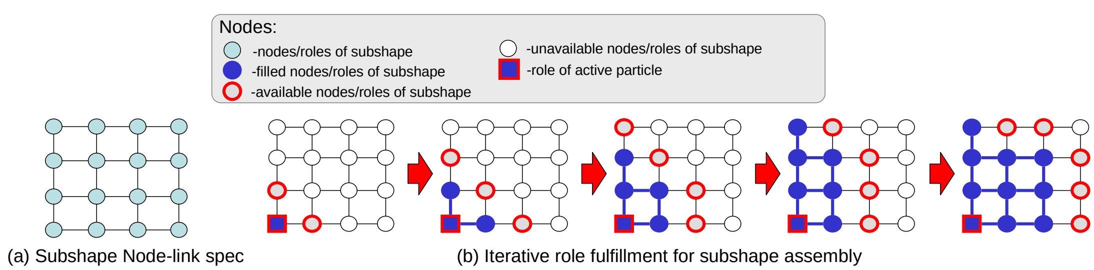


Figure 10: Example of Iterative Subshape Assembly

When a set of available roles is selected, the algorithm selects the particles to fill these roles based on a probabilistic assignment algorithm. This algorithm considers all available reactive particles (reactive only – since the active particles have already been selected in the first iteration) that are controlled by the active particle. First, we calculate a set of values d_{kj} for each pair of available role in the shape temporal plan $j \in V_D$ and a reactive particle $k \in V_C$. We do this based on the current x_k, y_k, z_k position of the active particles that already have assigned roles (nodes of the subplan) that have links with node $j \in V_D$ in the subshape G_D^r. We calculate the average distance of the active particle $k \in V_C$ to all active particles that already are assigned the roles from the subshape G_D^r (determined by role mapping matrix s_{kj}):

$d_{kj}(r) = \frac{1}{\sum_{i \in V_D} e_{ij}^D} \sum_{m \in V_C} \sum_{i \in V_D} e_{ij}^D s_{mi} u_{ri} \left((x_k - x_m)^2 + (y_k - y_m)^2 + (z_k - z_m)^2 \right)$. Alternatively, we could have calculated the exact position of where the new role should be located based on already filled roles (reactive particles) and the lengths and orientations of the connections.

We then minimize the objective equal to the summation of distances $\sum_{k \in} d_{kj}(r) s_{kj}$, which is equivalent to the *assignment problem formulation*. However, the assignment solution, while optimal at the time of the distances calculation, will quickly lose optimality since the particles move almost constantly. In addition, the assignment algorithm is of polynomial complexity, and we were looking for a linear complexity real-time solution. As the result, we decided to avoid using the assignment algorithms (such as *auction algorithm*) and use instead the randomized assignment, which is selecting a 0-1 matrix using the distances as probabilistic weights. Such an approach can be viewed as *multi-dimensional soft-max*. The resulting assignment was of linear complexity and provided solutions that were robust to particle movement.

Active Particle Command Network Design

To avoid decision-making confusion associated with the distribution of control, military organizations impose a *command structure* (i.e., *superior-subordinate* or *supported-supporting relations*) on their team members. One of the goals in creating a specific command structure is to match the induced superior-subordinate relationships among commanders with the coordination required to complete the mission. Different definitions of this matching lead to different formulations of the organizational command structure design problem.

For shape assembly controlled by the organization of active particles, we employ the same formalisms used for military command and control. The simplest command structure is a hierarchy with a single commander "root" active particle and all other active particles being subordinates to it. The problem with this setup is the overload of the monitoring and conflict resolution that will be imposed on the root active particle. Instead, we want to design the command structure among active particles to match the assembly coordination required among them. That is, the command structure must match the shape decomposition and plan designs.

We consider the situation when the coordination between any two active particles needed during shape construction requires the participation (e.g., monitoring, status update, approval, information passing, etc.) of all active particles involved in the corresponding command (superior-subordinate) path spanning two coordinating particles. That is, this accounts for

passing command-related information only via command structure network links, such that each active particle can communicate command-related messages only with its immediate superior/subordinate particles. The associated *coordination overhead* adds the extra load to each active particle involved in the decision cycle.

In our research, we limited the command networks topologies to a tree structure – enforcing a natural hierarchical relationships for commanders (each commander has at most a single superior, and only one "root" commander does not have a superior). Such a command structure sometimes is referred to as a tree. In this case, if $R_{s,r}$ defines the coordination requirements among active particles (e.g., this can be defined based on the subshape fusion required to be coordinated by the active particles, in which case $R_{s,r} = w_{link} \sum_{i,j} u_{si} u_{rj} e^D_{ij}$), then

$\boldsymbol{O}(m) = \sum_s \sum_{r>s} R_{s,r} \cdot \mathbf{1}(m \in \text{path in T from } s \text{ to } r)$ defines the coordination overhead for active particle m ("path" is found in the command hierarchy tree T, where a single path exists between any two nodes). The coordination overhead load is redundant and could potentially be avoided using a different command structure configuration.

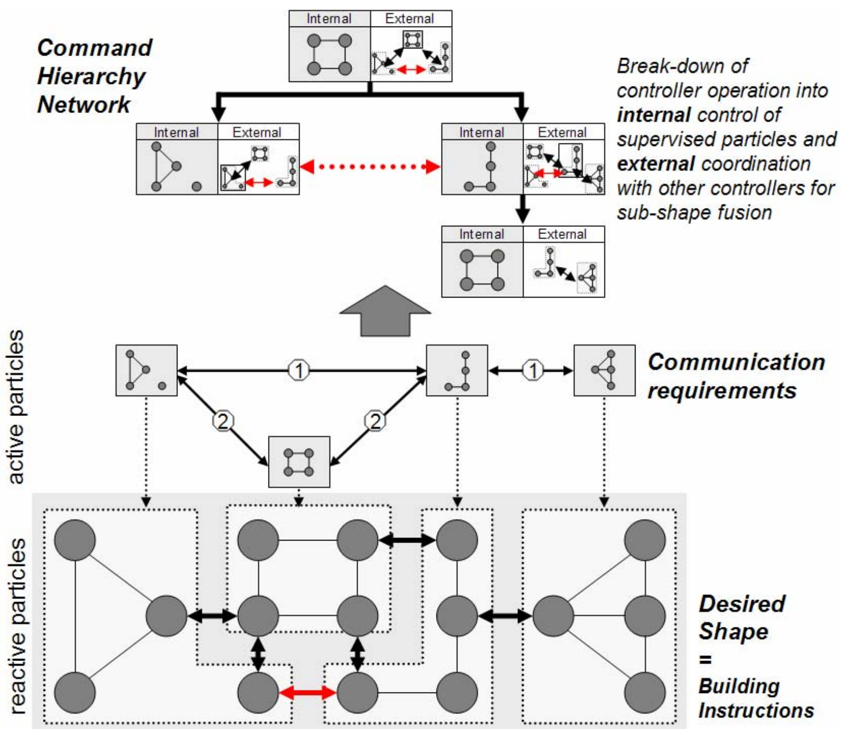


Figure 11: Example of Processes in Command Structure Induced by Shape Decomposition (Red arrows indicate the overhead coordination)

As the result, we are interested in designing the command hierarchies that minimize the total overhead coordination in the C2 organization computed as $\sum_m \boldsymbol{O}(m)$. Such command structure can be found suing the *minimum coordination cost tree* design algorithm which uses the max-flow (min-cut) approach to cluster the active particles and generate the tree structure. This algorithm has been used successfully for military C2 structure design in several previous projects for Navy and Army alternative organizational design analysis. We refer the readers to the following papers

for the detailed algorithm description (Levchuk et al., 2002; 2006). In the simulations we describe in later sections, we have experimented with several command structures. An example of how the shape decomposition interacted with command structure is shown in Figure 11.

Active Particle Communication Network Design

The communication network must also be designed to match the coordination among particles and corresponding communication of command-based and other types of information. This can be achieved using the optimal network design concepts to minimize the delays during communication (Levchuk et al., 2002; 2003; 3004; 2006). The reason for "designing" the communication structure is to avoid the overload that may occur when all one-to-one communication channels are open and to limit the memory required to store communicated information.

In our research, we also looked at the dependencies of the communication structure on the spatial positions of the active particles. The communication bandwidth could be a function of the distance between particles and the objects between them that become obstacles to wireless transmissions. We have also looked at the wired communication network design. For the latter, the constraints on the number of communication links and their bandwidth are even more essential than for the wireless communication networks.

Shape Plan Execution

Using the shape temporal plan variables $p_{sr} \in \{0,1\}$, we start defining the *in-degree* of the subshape G_D^r as $n(r) = \sum_s p_{sr}$. Initially, this variable is equal to the number of the subshapes that must be constructed immediately before G_D^r can be started. As subshape G_D^s is finished, we update the in-degree parameters $n(r) = n(r) - p_{sr}$. If $n(r) = 0$, the subshape G_D^r is activated for construction. The activation and parameter updates happen at active particles who are assigned the responsibility for subshape G_D^r activation.

Cognitive Particles Simulation Testbed Setup

To create an environment for testing, validating and comparing the programmable assembly theories and algorithms, we developed the *Cognitive Particles simulation testbed*. The testbed allowed us to develop principles relevant for guided, reversible shape assembly formation. As we began design of the testbed, it became clear that it would be helpful to be working in 3D with realistic physics. We therefore extended our intended Java-based AnyLogic prototype modeling environment from XJ technologies to include a package called Irrlicht 3D (for 3D modeling and visualization) and a package called Newton Dynamics (for physics simulation). These packages added 3-dimensional representation, collision detection and physics-related behaviors.

The high-level architecture design for the testbed is shown in Figure 12. The testbed consisted of four main components:

- The **Shape Object Specification** component was implemented as a database holding the node-link specifications of desired shapes and the shape assembly plans (shape decomposition and temporal plans);

- The **Particle Simulation** component, based on the Newton Dynamics physics engine, provided a realistic constructive simulation of the particle environment with physics-based effects (collisions, gravity, friction, joints, etc.) and tactical instruction execution (force-based particle shake, movements of the particles to create joints, collision analysis, callback for execution events, etc.).
- The **Control Simulation** component implemented the C2 design and distributed C2 assembly construction functionality. C2 structure design algorithms can be used for designing command, communication, and control networks of active particles. The user can also specify their own C2 structures. The actions for active particles to construct their sub-shapes and coordinate subshape fusion were defined using iterative role selection algorithm. The instructions were communicated and coordinated using the active particle C2 structures (command, communication, and control networks).
- The **User Interaction** component had 2-D viewer & controller (2-D layout, simulations and manual interaction controls, and measures), and 3-D viewer (3-D rendering of assembly based on Irrlicht engine).

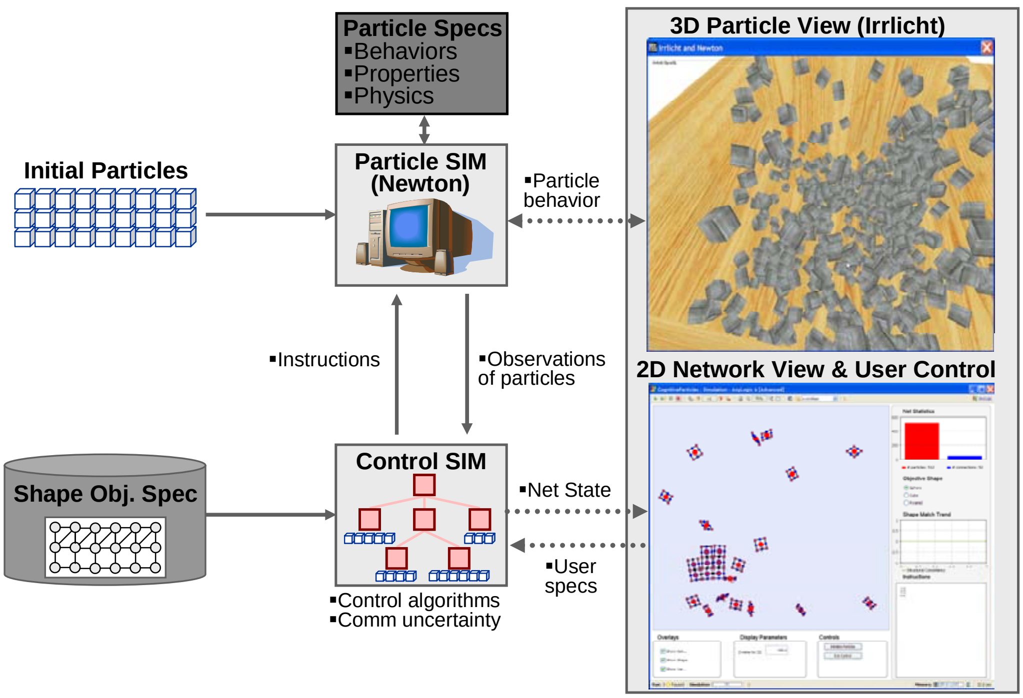


Figure 12: Cognitive Particles Testbed Components

We implemented the elementary particles as 3-D cube shapes with additional state representations. We have modified the particle class to store the information about its physics, visualization representations, organizational responsibilities, and connection properties. Command, communication and control networks were specified using directed graph data structures. These C2 networks were instantiated and updated over time using publish-subscribe mechanism; these networks enabled information routing and message queuing. The instruction messages to create/delete particles, change their states (e.g., color), and create/delete joints among particles were passed from Control SIM to Particle SIM and queued for execution. As the

result, the simulation provided a platform with on-line dynamic shape assembly visualization (Figure 13).

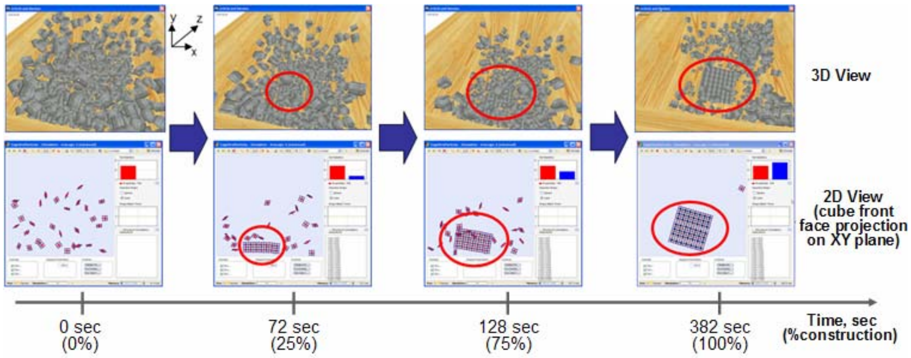


Figure 13: Example of Cube Shape Assembly in Cognitive Particles Testbed

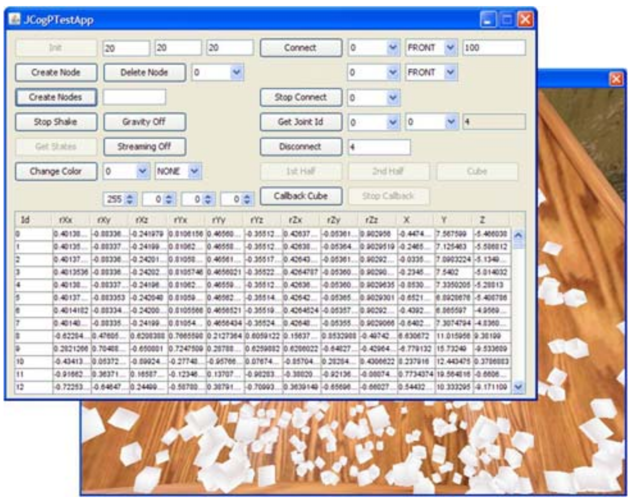


Figure 14: Cognitive Particles Test Application

In addition to the abovementioned components, we have developed a Cognitive Particles Test application (Figure 14) to study local particle behaviors, analyze API calls, study and improve performance of physics and graphics engines' functionality, analyze callbacks and particle spatial positioning, etc. Some of the implemented the functionality that was then used in the cognitive particles testbed is described below:

- *Generation of particles:* the application GUI allowed the user to generate the sets of needed size of active and reactive particles in the bucket.
- *Particle shake:* this functionality mimicked the physical shaking of the cube, as could be done in real world, by applying random forces of small value to the particles.
- *Gravity control:* the user is able to turn the gravity on and off. This was particularly useful to understand the particle mobility constraints.
- *Particle coloring:* we have implemented several coloring options – coloring a specific particle, or coloring the set of particles with diffused color effects based on the length of the communication.

- *Joint formation:* particles could be selected to join with each other. The faces of the joints must be specified by the user. One of the particles then starts moving to the other using a force attraction (mimicking the magnetic attraction). On its path, the particle could collide with other particles, and would execute detouring when stuck against the obstacles (particles in the way, semi-built subshapes, etc.). When the particle is very close to its destination, the local behavior algorithm computes the rotation necessary to achieve final orientation, and the particle is "teleported" (quickly rotated and moved) to the destination position.

Experimental Analysis

Experimental Metrics

Evaluation of the effectiveness of different assembly models and processes requires a novel set of metrics. We have begun to develop these metrics and have conducted initial experiments using the following set:

- **Metric 1 --- Timeliness:** Time to complete execution of shape plan
- **Metric 2 --- Accuracy:** Differences between currently assembled object and desired shape plan
- **Metric 3 --- Resources:** Amount of assembly resources which represent the cost of control in terms of manufacturing the required components. We computed this metric as the number of parallel channels of execution, i.e. number of active particles performing commanders' roles in C2 particle organization)
- **Metric 4 --- Energy:** Energy expended by particles to execute the assembly, which represents the cost of control to maintain the execution process.

Experimental Hypotheses

The objective of the assembly research we have conducted in this project was to develop an automated intelligent assembly control framework, including a model and a testbed that would achieve a feasible and efficient construction of shapes of interest. We started our analysis with simple shapes – including cube, sphere, pyramid – and moved to more complex shapes (e.g., wrench) that could have functional components.

The main hypothesis of our research was that for efficient automated shape assembly, there needs to be a *match between shape decomposition, temporal plan, the C2 particle organization, and assembly metrics* (Figure 15). The notion of the "match" between these components is know as the *congruence concept* in military and socio-technical command and control organizational analysis (Levchuk et al., 2002; 2003; Kleinman et al., 2003). While possessing the same "intelligence" (algorithms for developing assembly instructions and communicating them to the physical world), the mismatch between these components might result in waste of resources or energy, and delays or failures of shape assembly.

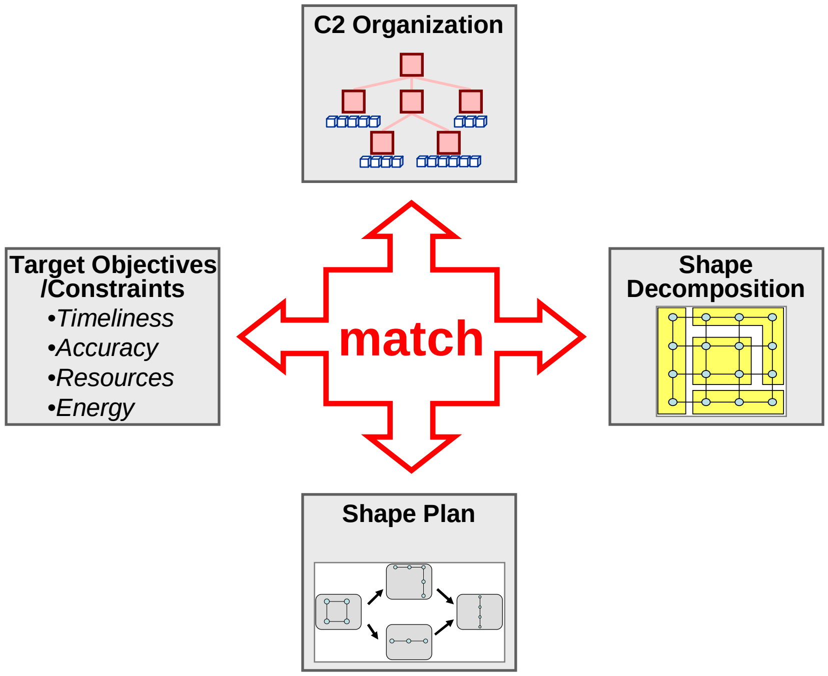


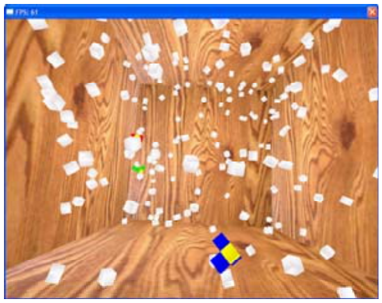
Figure 15: Conceptual Representation of the Correspondence among Assembly Elements

Run Setups

We have conducted several simulation runs, results for which are described below. For each run, the uncertainty model (probability of success of an instruction communicated to a subordinate) was fixed at 80%. For every run, there were fixed inputs for the shape decomposition and shape temporal plan. The number of active particles (commanders in C2 network) was fixed to equal the number of subshapes in the decomposition. We fixed the organizational command structure intentionally to be "flat" (i.e., a single superior commander active particle and all other active particles being subordinate to it). The communication structure was complete (all-to-all) with only geo-spatial restrictions (distances, obstacles). We thus compared the success of shape assembly under different shape decomposition, shape planning, and resources (number of active particles) constraints.

Example of Shape Assembly

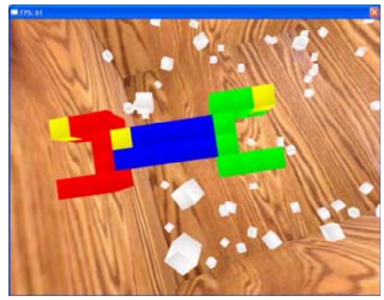
In Figure 16, we show an example of the shape assembly for a complex wrench object. The example illustrates how the shape assembly is happening over time by following the shape temporal plan and conducting distributed parallel subshape assemblies and subshape fusion activities by active particles. In this example, there are three active particles indicated with yellow color cubes in Figure 16. Each active particle is controlling 100 reactive particles. The active particles are constructing three subshapes indicated with distinct colors: red, blue, and green. White particles indicate reactive particles available as resources for the shape assembly if needed, but which are not currently part of the shape (that is, they have not been selected to fulfill shape node roles and not connected to the shape structure).



(a) Constructing subshapes in parallel

(b) Constructing subshapes in parallel – continued…

(c) Subshapes completed



(b) Subshapes fused – shape assembly completed

Figure 16: Assembling a Shape of Wrench (Figures *a* and *b* show each of the active particles constructing their own subshapes. Figure *c* shows that each active particle finished its subshape and is attempting to fuse its subshape with others'. Finally subfigure *d* shows the completed wrench)

Results and Discussion

Building a Cube of Cubes

In this first set of experiments, we automate the building of a 3x3x3 cube from a bucket of 100 cube-shaped particles. To better understand the influence of beginning shape construction with different numbers of commanders building different subshapes, we have three building plans:

1. a single, sequential construction of the 3x3x3 cube (led by one active particle)
2. construction of two sections of the cube first, followed by the fusion of these two sections (each led by an active particle, so the total is two active particles)
3. construction of each of the three 1x3x3 layers first in parallel, followed by their fusion (requiring three active particles in total).

These three decompositions are shown in Figure 17.

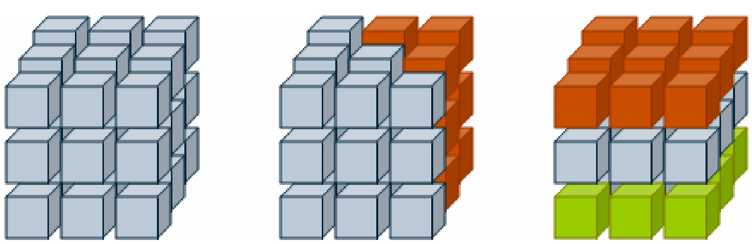


Figure 17: The three cube shape decompositions

For the experiments involving cube construction, the final shape itself required 3*3*3 = 27 cube particles. We therefore filled the particle bucket with 100 particles at the start of the simulation to ensure there were ample reactive particles from which to choose for the shape construction.

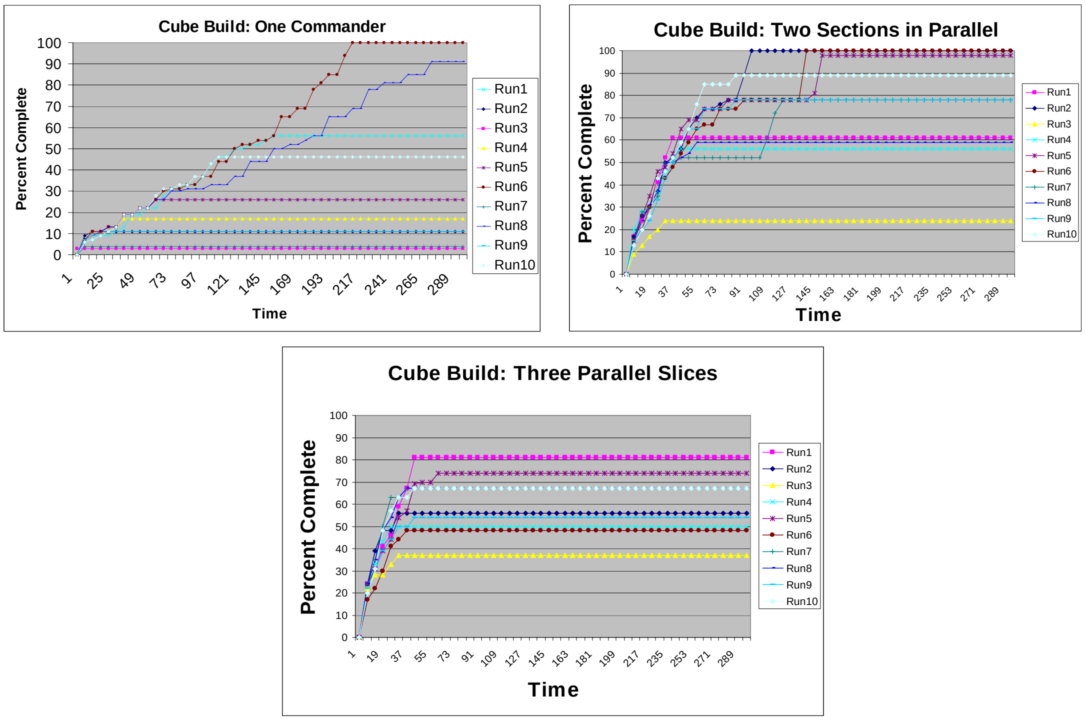


Figure 18: Cube construction results: 10 simulation runs each for the cube built serially with 1 commander (top left), with 2 sections built in parallel (top right), and with 3 sections (bottom)

We allowed the simulation trials to run for three hundred simulation time ticks each, and we recorded the progress in construction every 5 time ticks. Progress is reported in the percentage of the links required for the shape temporal plan that have been correctly accomplished by the report time. This cumulative measure then grows over the rest of the simulation trial. We begin each of these experiment trials with no gravity in the bucket, and the links are constructed one at a time. The results for these three experimental shapes are given in Figure 18.

In the simulation runs for which the percent complete reached 100% (the top of each of the graph areas in Figure 18), the particles successfully constructed the entire goal shape (a 3*3*3 cube, for example, in the Figure 18 simulations). The simulation runs for which the percent completion asymptotes were lower than that had only partial success.

As is clear from these progress graphs, many of the shape-building efforts became “stuck” in mid-construction. This phenomenon most often occurred when a required particle was trapped underneath a partially constructed shape, or when a particle trying to connect to the shape caused the shape to be trapped against a wall of the bucket (see Figure 19, for example). This behavior, and the frustration of watching shapes stuck in a corner or along a bucket wall, has led the insight that a very necessary component of a particle construction environment will be the ability to command a particle to “disengage” or “stop trying to connect” in order to find a way to move out of the deadlocked situation. In the wrench construction experiments (discussed below) we tried introducing a constant extra force in the form of shaking the bucket to try to reduce this problem, and we met with some limited success in this attempt.

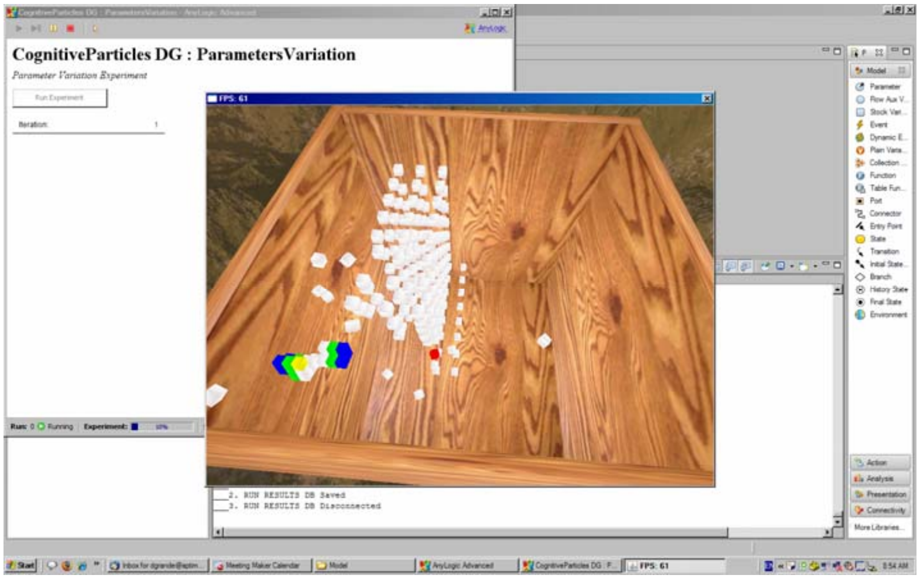


Figure 19: Two cube sections stuck against the wall during construction

In general, the parallel construction of the two sections that then fused into the cube had the greatest success. The parallel construction seems to have reduced the difficulty of constantly moving around a large, nearly-formed shape that were encountered in the single-active-particle constructions. The three-slice construction consistently began well, but the three slices were never able to correctly fuse into the cube. This particular problem leads us to believe that more study is required into the subshape fusion process, including analysis of the communications between active particles as well as simply the specification of the order and manner in which the multiple subshapes should try to connect.

Building a Wrench from Cubes

In order to attempt the construction of a more complicated shape, we developed three decompositions of a wrench for testing. These three are shown in Figure 20. The first is a wrench controlled by one active particle, and the construction is sequential. The second is a wrench decomposed into two equal halves, each controlled by one of the two active particles and then fused together. In the third, the three active particles each take one part of the wrench to construct in parallel and then fuse the subshapes together.

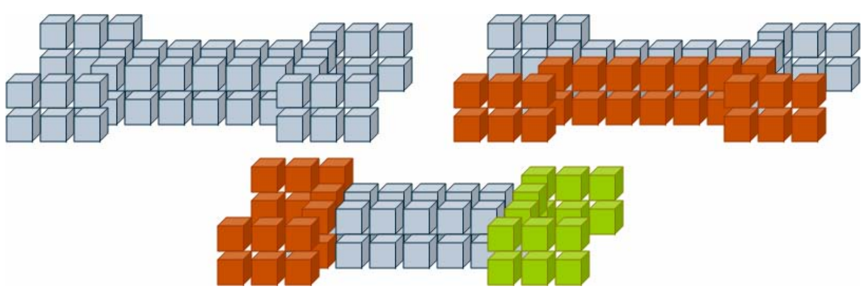


Figure 20: Wrench Shape Decompositions

For these experiments, because the wrench requires more particles than the cube (the plan has 52 particles), we initialize the bucket of particles with 200 instead of 100 particles. The results for

the experiment simulations with a single active particle (sequential shape building) are shown in Figure 21. Clearly, the single active particle had trouble constructing the 52-particle shape plan, and in most of the runs, the construction process was stalled by a particle or the subshape becoming stuck (as discussed above in the Cube Construction results section). It does appear, however, that Run 8 might have completed construction if given more time, but there were still nearly 20% of the links to go when the simulation time ended.

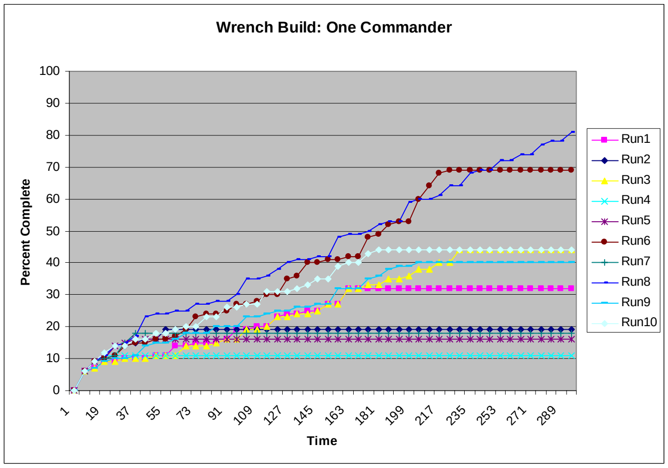


Figure 21: Wrench construction led by a single active particle

For the two-subshape wrench construction, we tried both having the active particles construct their respective subshapes in parallel (like we did with the 2-subshape cube above), and then we tried having them construct the two subshapes serially before fusing into the whole wrench. The idea was that although the serial construction should take longer than the parallel construction, it might be easier than serially constructing the entire shape with a single active particle (as was done in Figure 21). The results of these two sets of trials are shown in Figure 22.

In these two-subshape wrench construction trials, more of the shapes exceeded 50% construction than was possible in the single-shape wrench construction. This is a similar performance improvement to that seen in the cube construction above. The serial construction in these trials was allowed to run longer (400 simulation time ticks rather than 300), and the result was very similar performance for the sequential and parallel wrench builds.

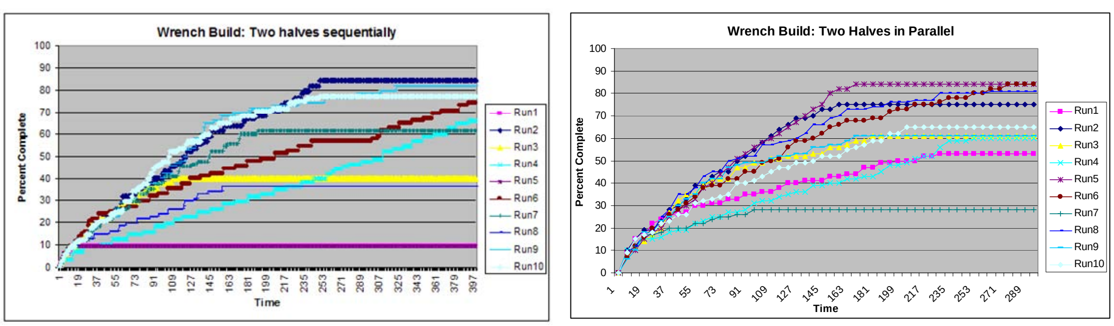


Figure 22: Wrench Construction via Two Subshapes: Results for 10 simulations runs each for building the wrench halves sequentially (left) or simultaneously in parallel (right) with construction Percent Complete (vertical axis) shown over time (horizontal axis)

Finally, we conducted experiments with the 3-subshape wrench construction. Here, we had the three active particles construct their subshapes in parallel and then attempted to fuse the three into the wrench plan. We ran an additional set of trials for this configuration in which we turned a "shake" force on in the bucket that caused the particles to be in continuous movement. The aim of this additional force was to try to prevent some of the occasions in which particles and subshapes would get stuck against the wall or each other. The results from these two 3-subshape simulation experiments are shown in Figure 23.

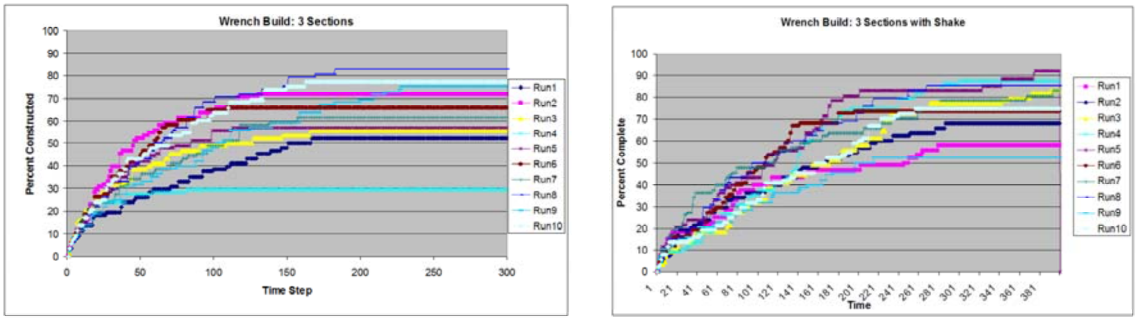


Figure 23: Wrench Construction via Three Subshapes

In general, although none of the wrenches were completely constructed, the original 3-subshape build had all but one of its wrenches complete more that 50% of the total shape. In addition to the fact that the three subshapes were building smaller pieces of the whole in parallel, the number of connections required to fuse the three subshapes is relatively limited. The success of this decomposition plan leads us to believe that there may be some benefit to this smaller number of inter-subshape connections that is worth further study.

The addition of the shake did appear to prevent much of the early construction stalling. We ran this simulation for an extra 100 time ticks (400 rather than 300), and there appears to have been aggressive link-forming for all trials through the first half of each run. This improvement argues for further study of the advantages of adding forces during the construction process. In particular, we would like to study whether policies for determining when such extra forces would be helpful and/or appropriate during the construction process.

Insights and Lessons Learned

A number of different lessons and have been learned as a results of the Cognitive Particles project:

- Need for managing online (object shape plan) versus offline (plan execution/shape assembly) tradeoffs
  - Offline is favorable due to limited computation power. *We can design the shape decomposition, the shape temporal plan, and the organizational structures that could potentially result in optimized assembly.*
  - Online is favorable due to dynamic, unpredictable sequence of events. *When the shape construction process starts to fail (which can happen due to failures in communication or instruction execution, particle mobility constraints, subshapes being stuck and not able to fuse, etc.), the old plan can no longer produce successful control. As the result, the plan needs to be changed – and the control*

process (shape assembly execution) must be adjusted online. However, complete distributed plan redesign with limited computational power may be infeasible.

- Benefits of decomposition: *Objects can be divided into separate components that can serve as an agreed method of distributing work for parallel execution.*
- Benefits of distributing command and control
  - More resilient to failures: *no single point of failure, information is already distributed therefore the environment is conducive to distributing responsibilities.*
  - Faster construction *via parallel channels of command and control due to multiple commanders (active particles).*
- Role allocation: Static allocations of active particles to subshape construction, allocation of reactive particles to be controlled by active particles, and allocation of reactive particles to shape plan node roles may become inefficient over time. *This is due to uncertainty in the future spatial collocation (location in the space) of the moving particles. availability of particles, and uncertainty of the success of communication. For example, the movements of particles completely changed the ability of the active particles to communicate with reactive particles, with allocated reactive particles being too far to the particles they needed to join with at the time of instruction execution.*
- Benefits of leveraging randomness – *While randomness can prevent some of the benefits of offline planning, by incorporating shake we were able to create more reliable object construction. Thus, we used the random movements to our advantage similarly to how the random walk algorithms allow the solution to avoid local optimums.*
- To achieve efficient construction it is necessary to balance interdependent forces of random shake, gravity, and magnetic propulsion.

Future Directions

In this section, we describe several directions of possible future research that we discussed during this project.

Designing an Adaptive C2 Organization for More Efficient Assembly Execution

There are two situations in which we need adaptability in the shape assembly execution. In the first situation, over time, the originally designed plan may become inefficient and need to be changed. In order to support plan redesign, a significant computational power is needed at individual particle level, and this may be infeasible at present.

In the second situation, as the shape temporal plan is executed over time, the organization that is most efficient to execute this plan at the beginning of the shape formation may not be as efficient at the middle of construction or at the end. The modeling described above for the C2 structure design takes the whole plan into consideration. Instead, we could analyze the plan temporal constraints, break the assembly into phases, and develop the organizational design that would enable changingC2 structure over time. One example of the savings is the number of the active particles, as then we do not need more active particles than the number of subshapes that could

be constructed in parallel (in current setup, the number of active particles is equal to the total number of subshapes). All active particles not assigned current subshapes would be idle; therefore, we can make an adaptive design by reassigning the subshape construction to available active particles over time. A similar, if not larger, improvement could be made by changing the command and communication structures among active particles to account for the changing coordination due to subshape fusion. These considerations could be incorporated ("compiled") into the C2 network offline.

Controlling Competitive Assemblies

Another interesting application and future direction of our work is developing control models for assembling the shape structure in the competitive environment. In military command and control simulations, this will represent the existence (of possibly multiple) opposing sides. In our situation, there could be multiple C2 networks coexisting with overlapping or conflicting goals. The control over reactive particles could then be competitive: that is, each active particle (or C2 network) may take away the control over the reactive particles. The applications of the competitive assembly can range from the medications that build healthy strands of bacteria to control of robotic forces.

Complex Assemblies: Heterogeneity and Uncertainty

In this research, we have experimented with homogeneous same-property particles and joints. Our approach is extensible to heterogeneous particles with different types of connections. This would allow, as discussed in introductory sections, to build objects with different material properties in its parts (e.g., a baseball with different layers, a wrench with soft-feel handle layer and steel ends, etc.). We could also incorporate different physical and chemical connection types, and more realistic constraints on the connection success and observations (knowledge that connections formed or got broken, knowledge of the location and state of reactive particles, knowledge of the spatial distribution of the active particles to be used for communication, etc.).

Constructing Adaptive Objects

Some of the objects may need to be adaptive to the situation. For example, imagine assembling the wrench which must adapt to the task (bolt). In this case, the active particles which are part of the wrench need to understand the tasks that the wrench may execute, and change the shape accordingly. This requires modeling the observation of the particles more comprehensively (active particles will provide observations about surrounding of the shape to learn the size of the bolt that the wrench is about to be applied to), as well as modeling the resizing the shape structure potentially through mechanical or structural means. In the latter case, this might require assembly and disassembly "on-the-fly" (during task execution by the constructed object) of some of the subshapes. One such example is illustrated in Figure 24.

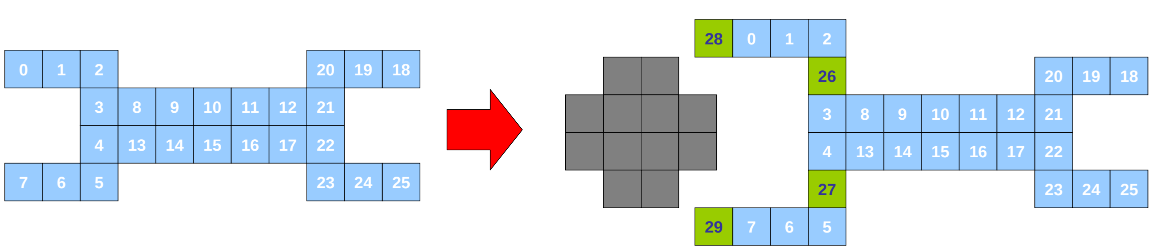


Figure 24: Adaptive Object Assembly (green squares indicate particles that must be added to adjust the wrench size)

Obtaining Feedback from the Active Particles

Another application of our work is in the feedback that active particles may provide about the observed structure of the currently assembled object shape. This is especially useful for material property assessment and diagnostics of the structural topology of unknown substances.

Programmable Ensembles

Programmable ensembles allow for connections that are beyond the physical, enabling the connected particles to have set distances between them or even have the connections be more abstract like that of communication or influence. Figure 25 shows a simple example of a programmable ensemble in which particles coordinate to form an expanded spatial cube. Mesoparticle-based programmable ensembles will generally exist in a liquid or a gas. As with programmable matter, the objective will be for the programmable ensembles to perform a mission, such as to create a shape with specific properties or to exhibit specific behavior like strengthening or disrupting existing structures in the environment.

In addition to modeling new types of connections, programmable ensembles also takes on a generalized notion of what a "particle" can represent, thus enabling new types of members to the ensemble. In previous work related to programmable matter, the aim was concentrate only on particles at the mesoparticle or even nanoparticle level. With programmable ensembles, the aim is to incorporate a wide variety of particle types that can be loosely connected, as seen in Figure 26. This opens up a wide variety of potential applications at for a wide variety of particle types. For example particles smaller than mesoparticles, there are a variety of medical diagnosis and repair procedures that would be enabled by intelligent behavior of the ensemble; for mesoparticles, one can imagine oil pipeline monitoring and repair or the formation of sails or wings that let just the right amount of air flow through them to maintain lift or velocity parameters, and one can even imagine moving solid ensemble objects that can temporarily disassemble to pass "through" (but really around) obstacles; if the particles are unmanned vehicles, the ensemble could adapt specific, intelligently adapted temporal or spatial patterns. Furthermore, troop formations of a human organization can also be viewed as a programmable ensemble that can be controlled and directed over time.

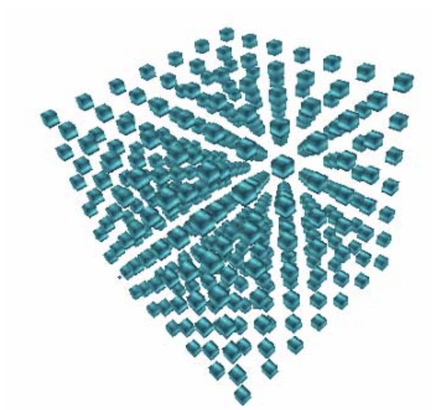


Figure 25: A programmable ensemble of particles forming a spatial cube

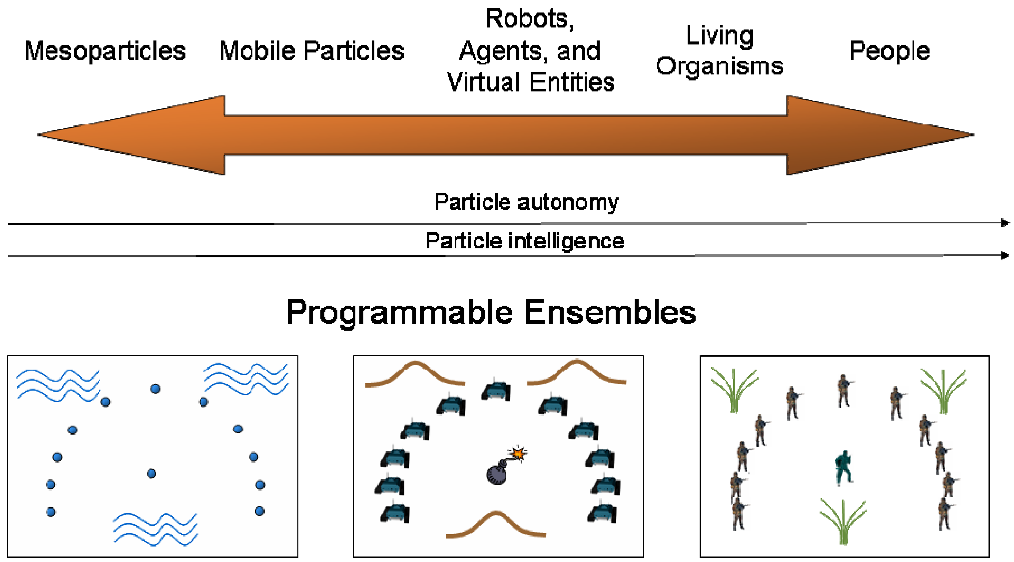


Figure 26: The spectrum of particles types in Programmable Ensembles

The addition to the freedom and flexibility that programmable ensembles brings, allows the ensemble to become much more adaptable. However, the issue of maintaining robustness in ensemble performance becomes much more salient. New and more difficult challenges arise. The non physical connections are more difficult to track the progress of and maintain. Since tasks are not physically constrained to those with adjacent neighbors, coordination itself becomes harder. Also the lack of adjacency makes it much more difficult or impossible to communicate. Furthermore, a key challenge to programmable ensembles is that this robustness must be achieved in general manner so that a programmable ensemble remains robust through many different missions (changing goal shapes).

One source of inspiration on how to approach this robustness is to consider analogs from nature. For example, ant colonies are a kind of natural ensemble. They can survive under a variety of environmental and other threats, though this is not necessarily true of individual ants. In the same way that we might ask what kind of attrition an ant colony might sustain and still remain an ant colony, we can ask what kind of attrition or other adverse events a Programmable Ensemble might sustain and still remain that same Programmable Ensemble. And in the case of Programmable Ensembles, we can go even further and ask how we might design the Ensemble's composition and mission to optimize its survivability in the face of adverse conditions.

References

Levchuk, G. M., Levchuk, Y. N., Luo, J., Pattipati, K. R., & Kleinman, D. L. (2002). Normative design of organizations - Part II: Organizational structure. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 32(3), 360-375.

Levchuk, G. M., Feili Y., Pattipati, K. R., & Levchuk, Y. (2003). From hierarchies to heterarchies: Application of network optimization to design of organizational structures. Proceedings of the 8th International Command and Control Research and Technology Symposium, Washington, DC.

Levchuk, G. M., Kleinman, D. L., Ruan, S., & Pattipati, K.R. (2003). Congruence of human organizations and missions: Theory versus data. Proceedings of the 8th International Command and Control Research and Technology Symposium, Washington, DC.

Kleinman, D. L., Levchuk, G. M., Hutchins, S. G., & Kemple, W. G. (2003). Scenario design for the empirical testing of organizational congruence. Proceedings of the 8th International Command and Control Research and Technology Symposium, Washington, DC.

Levchuk, G. M., Meirina, C., Pattipati, K.R., & Kleinman, D.L. (2004). Normative design of project-based organizations: Part III - Modeling congruent, robust and adaptive organizations. IEEE Trans. on SMC: Part A: Systems and Humans, 34(3), 337-350.

Levchuk, G. M., Yu, F., Levchuk, Y., & Pattipati, K. R. (2004). Networks of decision-making and communicating agents: A new methodology for design and evaluation of organizational strategies and heterarchical structures. Proceedings of the 9th International Command and Control Research and Technology Symposium, San Diego, CA.

Levchuk, G., Levchuk, Y., Weil, S. A., & Pattipati, K. (2006). PERSUADE: Modeling framework for the design of modular Army organizations. Proceedings of the 2006 Command and Control Research and Technology Symposium, San Diego, CA.

